CSE422(Summer 2023)
Artificial Intelligence

Topic: Diabetes Prediction

Group Members

Hasin Arman Prokriti (20201092)

Mashrur Ahmed Utsho (20201072)

Arnob Majumder (20301089)

Iftekhar Al-Mahmud (20201120)

Introduction

In this project, we embark on developing multiple **machine learning models for the prediction of Diabetes**, one of the very common diseases around the world. We aim to address the challenge of accurately determining whether one has diabetes or not. Our project seeks to provide a data-driven approach to solve the problem of uncertainty and subjectivity in diabetes finding.

Dataset Description

- **SourceLink:**
  Diabetes Health Indicators Dataset | Kaggle

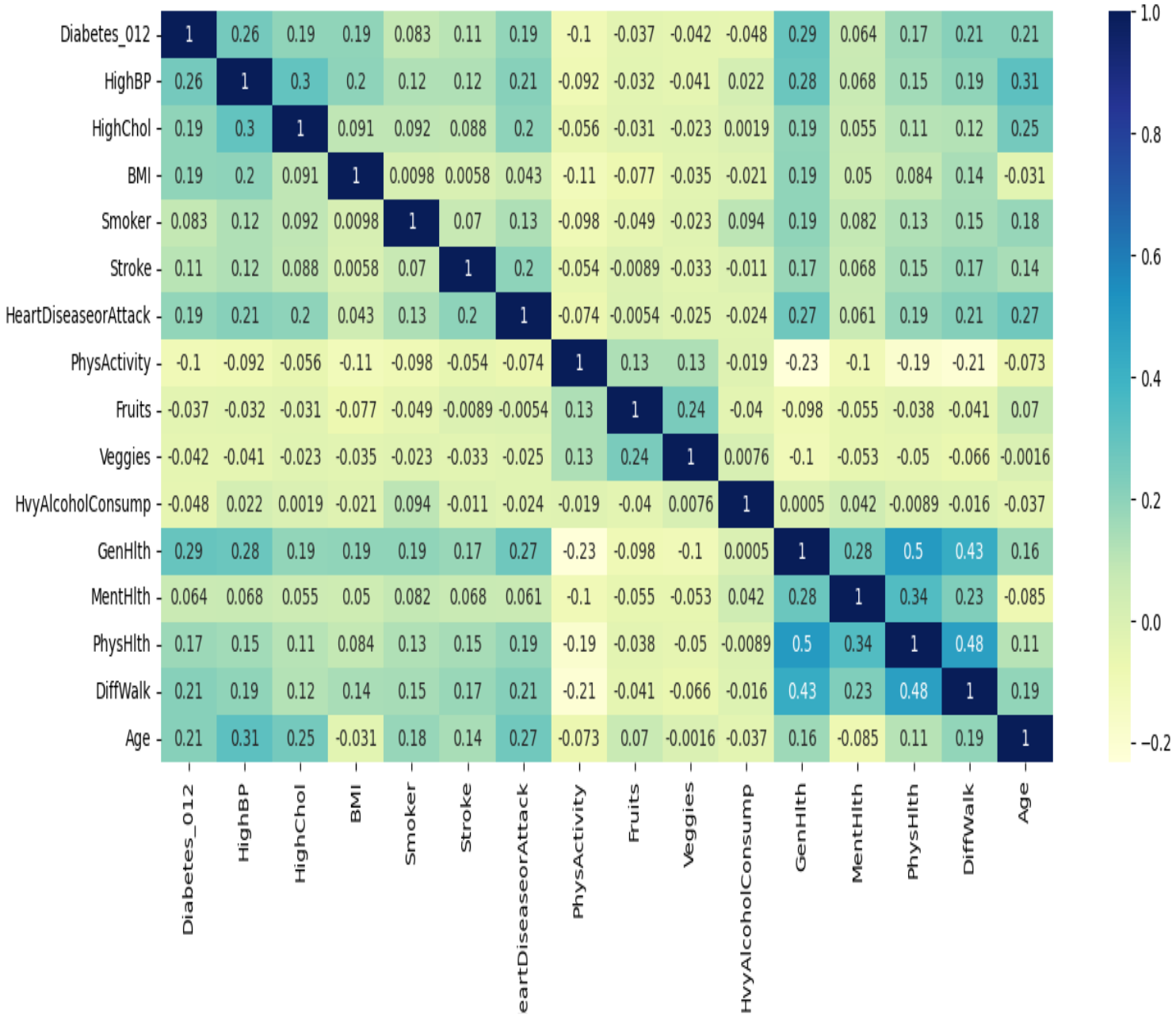
- **Dataset Description**:
  - In this dataset there are initially 21 features but after removing features that are non related to the diabetes detection it turns into 16.

  - This is a classification problem. Because in this project, We are classifying to predict whether one has diabetes or not depending on various factors. That's why this is a classification problem.

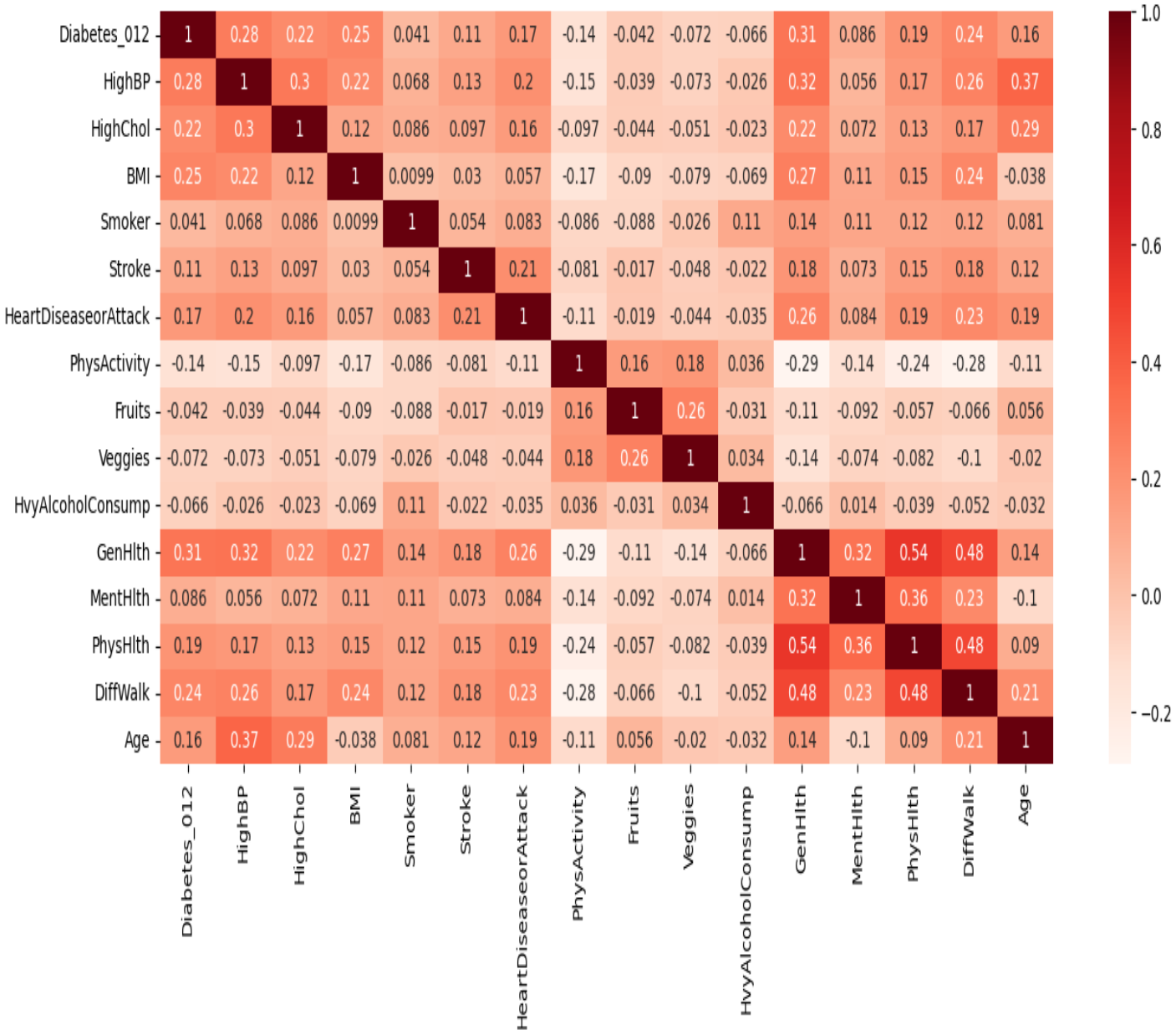
  - In our dataset, we have both quantitative and categorical features. Some features like Stroke and others are categorical on the other hand, BMI and some other features are quantitative.

Correlation of all the features:

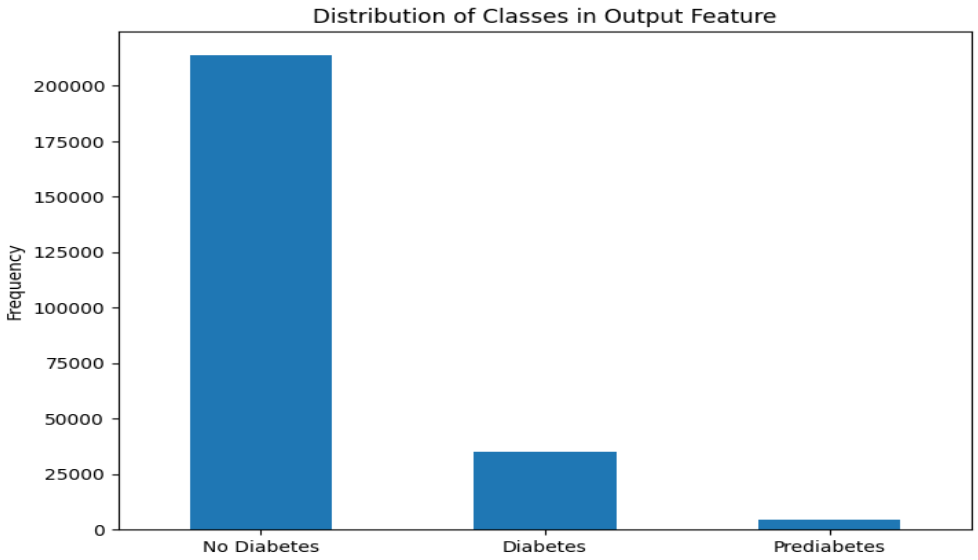
Male Correlation



Female Correlation



Imbalance dataset



Here we can see that there are 3 output classes. However, the distribution of data is imbalanced. To fix this we used random oversampling instead of undersampling because undersampling reduces the data, which reduces the accuracy of the model.

Fixing imbanalce by random oversampling

```
[ ]  1  from imblearn.over_sampling import RandomOverSampler
     2  # Apply random oversampling
     3  ros = RandomOverSampler(random_state=42)
     4  male_x, male_y = ros.fit_resample(male_x, male_y)
```

Dataset PreProcessing

- **Fault:**
  - Unrelated features
  - Male and female data together

- **solution:**
  - Changing the categorical values into boolean values

  - Dropping the unrelated features such as: income, education, insurance etc. as these features are not related to predicting diabetes of a patient.

  - Separating the data into male and female data. This is done because the occurrence of diabetes can be different for male and female; which would have led to misprediction of outcome.

Dataset Splitting

Total dataset was initially divided into two dataset based on male and female. Then these data are further splitted into train and test sets with the ratio of 70% for the train set and 30% for the test set.
While splitting the dataset, the data were stratified by the outcome feature: 'Diabetes_012'. This is done to make sure that the distribution of outputs are properly done so that each split has data of each different output classes.

Model training & Testing

- **KNN**

  The k-nearest neighbors algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. While it can be used for either regression or classification problems, it is typically used as a classification algorithm, working off the assumption that similar points can be found near one another.

- **Random Forest**

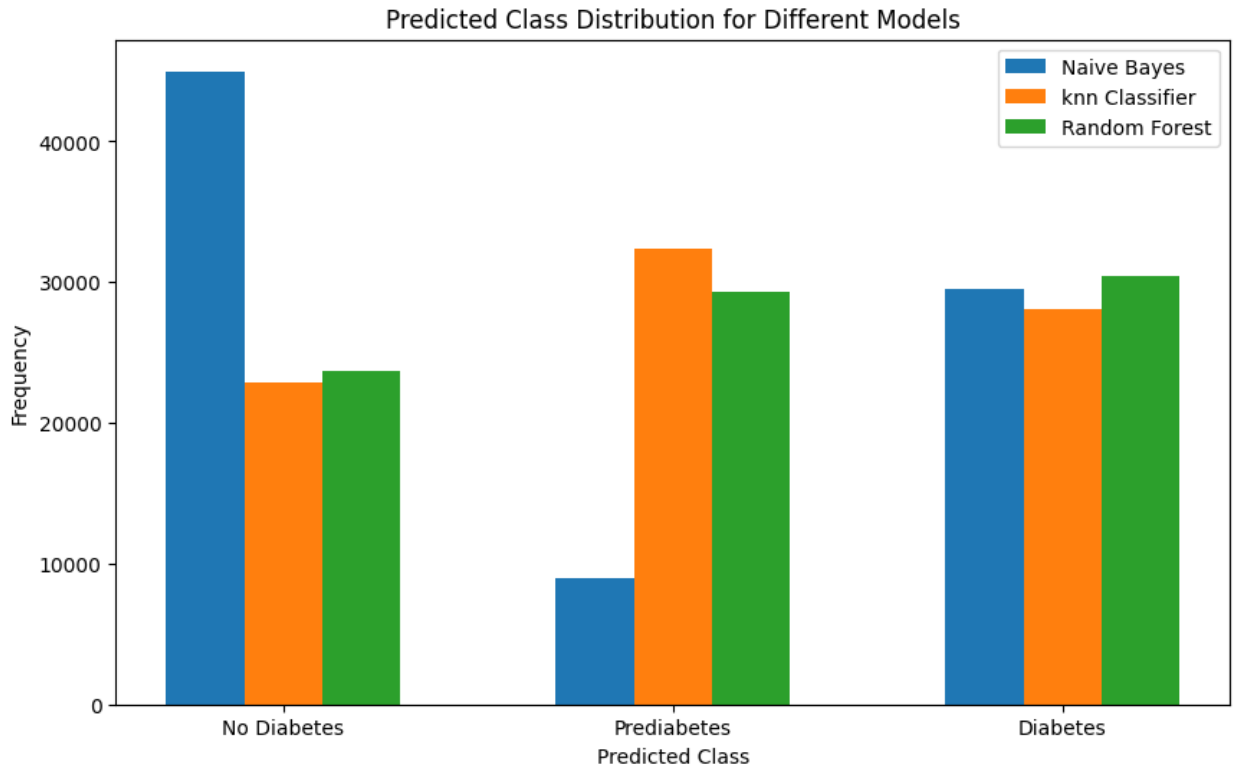
  Random forest is a supervised learning algorithm. The "forest" it builds is an ensemble of decision trees, usually trained with the bagging method. The general idea of the bagging method is that a combination of learning models increases the overall result. One big advantage of random forest is that it can be used for both classification and regression problems, which form the majority of current machine learning systems.

- **Naive Bayes Classifier**

  Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable. The different naive Bayes classifiers differ mainly by the assumptions they make regarding the distribution of $P(Xi \mid y)$. In spite of their apparently over-simplified assumptions, naive Bayes classifiers have worked quite well in many real-world situations, famously document classification and spam filtering. They require a small amount of training data to estimate the necessary parameters.

Comparison Analysis

Barchart of all the model charts



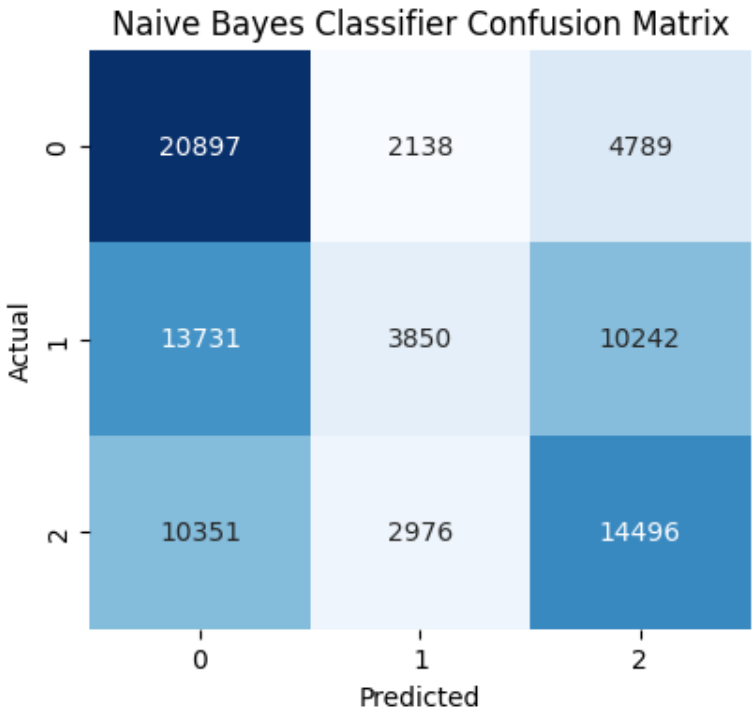
precision, recall comparison of each model

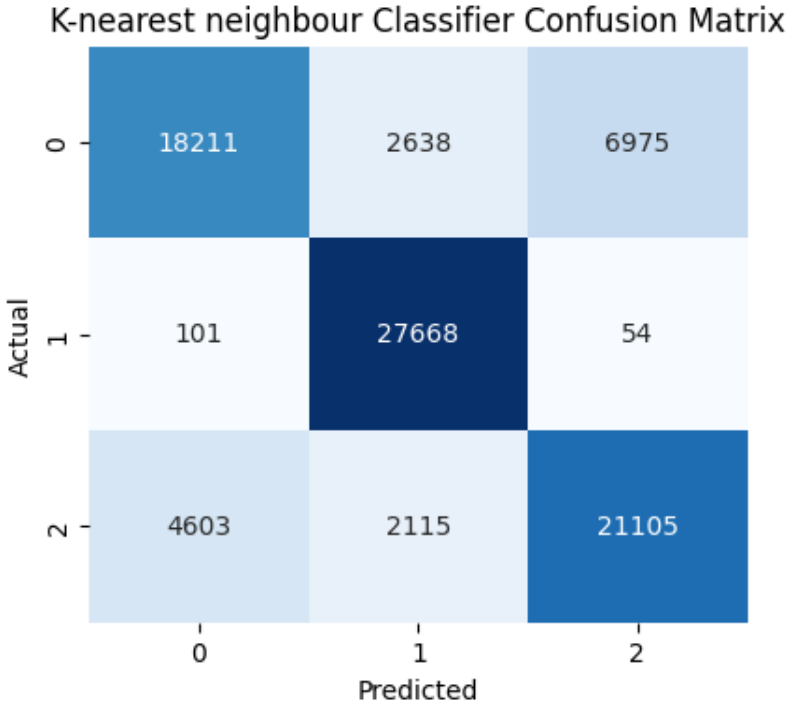
Naive Bayes Precision and Recall		
Class	**NB Precision**	**NB Recall**
No Diabetes	0.46459	0.75104
Prediabetes	0.4295	0.13837
Diabetes	0.49094	0.52101

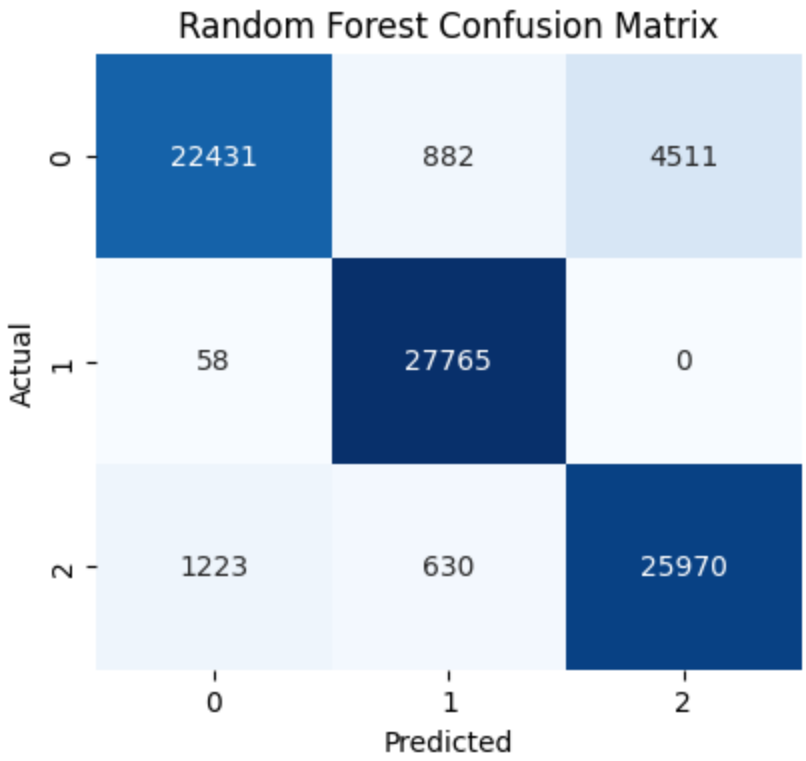
Knn Precision and Recall		
Class	**KNN Precision**	**KNN Recall**
No Diabetes	0.79472	0.65451
Prediabetes	0.8534	0.99443
Diabetes	0.75016	0.75855

Random Forest Precision and Recall		
Class	**Random Forest Precision**	**Random Forest Recall**
No Diabetes	0.94598	0.80617
Prediabetes	0.94836	0.99792
Diabetes	0.85201	0.9334

confusion matrix for each model







Conclusion

In conclusion, the results of our diabetes prediction machine learning project reveal significant variations in precision across different models. If we look at the precision and recall table for all the models we can see that the Naive Bayes model demonstrated comparatively lower precision in its predictions. On the other hand, the K-nearest neighbor model exhibited a notably higher precision, suggesting its proficiency in identifying true positive cases. However, the best performing model in our project is the Random forest model. These observations are further proved by the confusion matrix outputs.